Advanced HB

1. Device should support up to 5 types of HB packets for each monitor port in each segment.
2. Segment should switch to expired state if number of working segment decreased below the minimum working number (selected by command)
3. The segment should restore inline state when the number of working members reaches the minimum working number.
4. Segment should be able to return to inline mode manually or automatically.
5. Each HB packet for each segment can have different content except MAC addresses.
6. HB packet mode:
7. HB send from MON0, and receive response by MON1
8. HB send from MON1, and receive response by MON0
9. Bidirectional: send from each port and receive response from the counterpart port
* Segment will switch to expired state if one of direction will be failed (selected by command)
* Segment will switch to expired state if both of direction will be failed (selected by command)
1. Response content must be the same for any type of inline application processing.
2. New and old CLI commands needed for multiple HB-type feature (see in appendix A commands help and examples):
3. del\_hb - delete HB packet
4. set/get\_hb\_params - set/get HB packet type and other parameters
5. set/get\_hb\_state - set/get HB packet state
6. set/get\_min\_work\_members - set/get minimal number of HB-types working members for segment
7. load\_hb\_pkt - load HB packet content, should be compatible with HB type.
8. get\_hb\_pkt - display HB packet
9. get/set\_hb\_src\_mac - set HB packet source MAC for current segment
10. get/set\_hb\_dst\_mac - set HB packet destination MAC for current segment
11. set\_default\_hb\_mac - set HB packet default source and destination MAC for current segment
12. set\_default\_hb\_pkt - set default HB for current segment
13. IP used in HB packets must not be used in any other traffic.
14. MAC-type HB packet is not compatible with IP-type and VLAN\_IP-type HB packets.
15. Only one MAC-type HB packet can exist for one segment.
16. Each IP/VALN\_IP-type packet should have source and/or destination IP different from other packets.
17. HB packets install order:
* del\_hb (used only if needed to prevent using incompatible HB types.
* load\_hb\_pkt

 - set\_hb\_params

1. HB packet can be entered by binary or text file
2. HB packet file name must contain prefix “hb\_” and suffix “bin” or “txt”
3. HB packet file name can contain 8 – 20 characters including prefix and suffix.

Allowed following characters:

A-Z, a-Z, 0-9, '\_', '.', '-', '@', ':', '>', '+', '=', ','

1. HB packet size:
2. Binary file: 24-1024
3. Text file: 72-3454
4. HB packet file should not contain HB packet checksum
5. HB packet text file example:

00 50 c2 3c 60 00 00 50 c2 3c 60 01 81 37 ff ff

00 30 00 00 00 00 40 04 ec a2 c6 13 01 02 c6 13

01 01 00 00 00 00 00 00 00 00 00 00 01 02 03 04

05 06 07 08 09 0a 0b 0c 0d 0e 0f 00 10 20 30 40

65 66 67 66

1. Adding new HB packets feature will decrease filters number from 244 to 396

# HB packets type



## MAC HB packet

MAC HB packets have different MAC addresses for each segment.

MAC addresses can be changed by command

set\_hb\_dst\_mac: MAC1

set\_hb\_src\_mac: MAC2

HB packet that will be send from MON0 port

|  |  |  |
| --- | --- | --- |
| Destination MAC | Source MAC | payload |
| MAC1 | MAC2 |  |

Response:

|  |  |  |
| --- | --- | --- |
| Destination MAC | Source MAC | payload |
| MAC2/MAC1 | MAC1/MAC2 |  |
| Used for detecting HB |  |

HB packet that will be send from MON1 port

|  |  |  |
| --- | --- | --- |
| Destination MAC | Source MAC | payload |
| MAC2 | MAC1 |  |

Response:

|  |  |  |
| --- | --- | --- |
| Destination MAC | Source MAC | payload |
| MAC1/MAC2 | MAC2/MAC1 |  |
| Used for detecting HB |  |

## IP HB packet

MAC HB packets have different MAC addresses for each segment.

IP HB packets have different IP addresses for HB type in segment.

MAC addresses can be changed by command

set\_hb\_dst\_mac: MAC1

set\_hb\_src\_mac: MAC2

set\_hb\_dst\_ip: IP1

set\_hb\_src\_ip: IP2

HB packet that will be send from MON0 port

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Destination MAC | Source MAC | type | IP header | Payload |
|  | Source IP | Destination IP |  |
| MAC1 | MAC2 | 08 00 |  | IP2 | IP1 |  |  |
|  |  |  |  |  |

Response:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Destination MAC | Source MAC | type | IP header | Payload |
|  | Source IP | Destination IP |  |
| MAC2 | MAC1 | 08 00 |  | IP1/IP2 | IP2/IP1 |  |  |
|  |  |  |  | Used for detecting HB |  |  |

HB packet that will be send from MON1 port

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Destination MAC | Source MAC | type | IP header | Payload |
|  | Source IP | Destination IP |  |
| MAC2 | MAC1 | 08 00 |  | IP1 | IP2 |  |  |
|  |  |  |  |  |

Response:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Destination MAC | Source MAC | type | IP header | Payload |
|  | Source IP | Destination IP |  |
| MAC1 | MAC2 | 08 00 |  | IP2/IP1 | IP1/IP2 |  |  |
|  |  |  |  | Used for detecting HB |  |  |

## IP with VLAN HB packet

MAC HB packets have different MAC addresses for each segment.

IP HB packets have different IP addresses for HB type in segment.

MAC addresses can be changed by command

set\_hb\_dst\_mac: MAC1

set\_hb\_src\_mac: MAC2

set\_hb\_dst\_ip: IP1

set\_hb\_src\_ip: IP2

set\_hb\_vlan: VLAN (should not be internal Broadcom VLAN – default: 1, can be changed by set\_int\_vlan)

HB packet that will be send from MON0 port

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Destination MAC | Source MAC | 802.1Q | type | IP header | Payload |
|  | Source IP | Destination IP |  |
|  |  | 8100 |  |  | VID | 08 00 |  |  |  |  |  |
| MAC1 | MAC2 |  |  |  | VLAN |  |  | IP2/IP1 | IP1/IP2 |  |  |

Response:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Destination MAC | Source MAC | 802.1Q | type | IP header | Payload |
|  | Source IP | Destination IP |  |
|  |  | 8100 |  |  | VID | 08 00 |  |  |  |  |  |
| MAC2 | MAC1 |  |  |  | VLAN |  |  | IP1/IP2 | IP2/IP1 |  |  |
|  |  |  |  |  |  |  |  | Used for detecting HB |  |  |

HB packet that will be send from MON1 port

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Destination MAC | Source MAC | 802.1Q | type | IP header | Payload |
|  | Source IP | Destination IP |  |
|  |  | 8100 |  |  | VID | 08 00 |  |  |  |  |  |
| MAC2 | MAC1 |  |  |  | VLAN |  |  | IP1 | IP2 |  |  |

Response:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Destination MAC | Source MAC | 802.1Q | type | IP header | Payload |
|  | Source IP | Destination IP |  |
|  |  | 8100 |  |  | VID | 08 00 |  |  |  |  |  |
| MAC1 | MAC2 |  |  |  | VLAN |  |  | IP2 | IP1 |  |  |
|  |  |  |  |  |  |  |  | Used for detecting HB |  |  |

Appendix A

**Note 1**: Almost each command contains [module:segment[hb\_id]] | [hb\_id] parameter:

* If module:segment was entered, but hb\_id parameter was not – command work with first HB packet for specified segment.
* If module:segment:hb\_id parameter was entered – command work with specified HB packet for specified segment.
* If only hb\_id parameter was entered – command work with specified HB packet for current segment.

**Note 2**: Swap parameters mean that this parameter is swapped by user application when send from MON0 or MON1 ports.

**del\_hb** – delete one HB type

del\_hb <[module:segment:hb\_id] | [hb\_id]>

 - delete HB for current or specified segment.

 hb\_id - HB type index (1 - 5).

 If module:segment:hb\_id parameter was

 entered - delete the specified HB

 of the specified segment.

 If only hb\_id parameter was entered - delete

 the specified HB of the current segment.

**Examples**:

del\_hb 1

del\_hb 1:1:3

**set\_hb\_params** - set HB type and parameters

set\_hb\_params <[module:segment[:hb\_id]] | [hb\_id]>

 <mac m0\_swap\_src\_mac m0\_swap\_dst\_mac m1\_swap\_src\_mac m1\_swap\_dst\_mac>

 <ip src\_ip dst\_ip m0\_swap\_src\_ip m0\_swap\_dst\_ip m1\_swap\_src\_ip m1\_swap\_dst\_ip>

 <valn\_ip vlan src\_ip dst\_ip m0\_swap\_src\_ip m0\_swap\_dst\_ip m1\_swap\_src\_ip

 m1\_swap\_dst\_ip>

 - set HB parameters for the current or specified

 segment.

 hb\_id - HB type index (1 - 5).

 m0/m1\_swap\_src\_mac (on|off) - set the location

 of src MAC in the RSP packet on mon1/mon0.

 m0/m1\_swap\_dst\_mac (on|off) - set the location

 of dst MAC in the RSP packet on mon1/mon0.

 vlan - HB vlan number,

 src\_ip - HB source IP,

 dst\_ip - HB destination IP,

 m0/m1\_swap\_src\_ip (on|off) - set the location

 of src IP in the RSP packet on mon1/mon0.

 m0/m1\_swap\_dst\_ip (on|off) - set the location

 of dst IP in the RSP packet on mon1/mon0.

**Examples**:

set\_hb\_params 1 mac off off off off

set\_hb\_params 2 mac off on off on

set\_hb\_params 1 ip 192.168.1.0 193.168.1.2 off off off off

set\_hb\_params 2 ip 192.168.2.0 193.168.2.2 off off off off

set\_hb\_params 2 vlan\_ip 12 192.168.2.0 193.168.2.2 off off off off

set\_hb\_params 1 vlan\_ip 12 192.168.2.0 193.168.2.2 off off off off

set\_hb\_params 3 ip 192.168.3.0 193.168.3.2 off off off off

set\_hb\_params 4 ip 192.168.4.0 193.168.4.2 off off off off

set\_hb\_params 5 ip 192.168.5.0 193.168.5.2 off off off off

**get\_hb\_params** - display parameters for HB packet

get\_hb\_params [module:segment[:hb\_id]] | [hb\_id]

 - show HB parameters for the current or specified

 segment.

 hb\_id - HB type index (1 - 5).

 If module:segment was entered, but hb\_id

 parameter was not - will show parameters

 of the first HB of the specified segment.

 If module:segment:hb\_id parameter was

 entered - will show parameters of the

 specified HB of the specified segment.

 If only hb\_id parameter was entered - will

 show parameters of the specified HB

 of the current segment.

**Examples**:

get\_hb\_params 1:1:1

HB 1 parameters:

 state: loaded, configured, used, working,

 type: VLAN\_IP,

 vlan: 11,

 swap mon0 src IP: off,

 swap mon0 dst IP: off,

 swap mon1 src IP: off,

 swap mon1 dst IP: off,

 source IP 192.168.11.11,

 destination IP 193.168.11.12.

**set\_hb\_state** - enable/disable HB state

set\_hb\_state <[module:segment[:hb\_id]] | [hb\_id]> on|off

 - set HB state for current or specified segment.

 By default, HB enabled after setting HB

 parameters.

 hb\_id - HB type index (1 - 5).

 If module:segment was entered, but hb\_id

 parameter was not - set the state for

 the first HB of the specified segment.

 If module:segment:hb\_id parameter was entered -

 set the state for the specified HB

 of the specified segment.

 If only hb\_id parameter was entered - set the

 state of the specified HB of the current segment.

**Examples**:

set\_hb\_state 1 off

set\_hb\_state 3 on

**get\_hb\_state** – display HB state

ERROR: wrong parameter value or its length ("dd")

get\_hb\_state [[module:segment[:hb\_id]] | [hb\_id]]

 - show HB state for the current

 or specified segment.

 hb\_id - HB type index (1 - 5).

 If module:segment was entered, but hb\_id

 parameter was not - will show the state of

 the first HB of the specified segment.

 If module:segment:hb\_id parameter was entered -

 will show the state of the specified HB

 of the specified segment.

 If only hb\_id parameter was entered - will show

 the state of the specified HB

 of the current segment.

**Examples**:

 get\_hb\_state

HB 1 state: MAC , loaded, configured, used, expired.

HB 2 state: NONE, loaded.

HB 3 state: NONE.

HB 4 state: NONE.

HB 5 state: NONE, loaded.

get\_hb\_state 1:1

HB 1 state: VLAN\_IP, loaded, configured, used, working.

HB 2 state: VLAN\_IP, loaded, configured.

HB 3 state: VLAN\_IP, loaded, configured, used, working.

HB 4 state: VLAN\_IP, loaded, configured, used, working.

HB 5 state: VLAN\_IP, loaded, configured, used, working.

get\_hb\_state 3:1

HB 1 state: VLAN\_IP, loaded, configured.

HB 2 state: VLAN\_IP, loaded, configured.

HB 3 state: VLAN\_IP, loaded, configured, used, expired.

HB 4 state: VLAN\_IP, loaded, configured, used, expired.

HB 5 state: VLAN\_IP, loaded, configured, used, expired.

**set\_hb\_min\_work\_members** - set minimal HB working members

set\_hb\_min\_work\_members [module:segment] all|1-5

 - set a minimum number of working HBs before

 transitioning current or specified segment

 to expired state.

 all - all HBs should work.

 With this value - changing the number of

 HBs will automatically correct this

 parameter.

 1 - 5 - With these values - changing the

 number of HBs will not change this parameter.

 If module:segment parameter was entered -

 set the number of minimum working members

 for the specified segment,

 otherwise - set the number of minimum working

 members for the current segment.

**Examples**:

set\_hb\_min\_work\_members all

set\_hb\_min\_work\_members 3

**get\_hb\_min\_work\_members** - displays minimal HB working members for segment

get\_hb\_min\_work\_members [module:segment]

 - show a minimum number of working HBs before

 transitioning current or specified segment

 to expired state.

 If module:segment parameter was entered -

 will show the number of minimum working members

 of the specified segment,

 otherwise - will show the number of minimum

 working members of the current segment.

**Examples**:

get\_hb\_min\_work\_members

HB minimum work members: 1.

**load\_hb\_pkt** – load content for HB packet

load\_hb\_pkt file\_name [ip root] [module:segment[:hb\_id]] | [hb\_id]

 - loads new hb packet,

 file\_name - 8-20 characters, prefix: hb\_

 file should contain HB context (ext: txt or bin)

 and be in the tftp root directory.

 ip - tftp server IP address,

 root - tftp root directory.

 hb\_id - HB type index (1 - 5).

 If module:segment parameter was not entered -

 the packet will be used for the first HB (hb\_id=1)

 of all segments.

 If module:segment parameter was entered,

 but hb\_id was not - the packet will be used for

 the first HB (hb\_id=1) of the specified segment.

 If module:segment:hb\_id parameter was entered -

 the packet will be used for the specified hb\_id

 of the specified segment.

**Examples**:

load\_hb\_pkt hb\_tcp.txt 192.168.0.2 tftpboot/hb\_new/IP 2

load\_hb\_pkt hb\_tcp\_unix.txt 192.168.0.2 tftpboot/hb\_new/IP 2

load\_hb\_pkt hb\_tcp.bin 192.168.0.2 tftpboot/hb\_new/IP 2

load\_hb\_pkt hb\_ping\_ip.txt 192.168.0.2 tftpboot/hb\_new/IP 2

load\_hb\_pkt hb\_vlan6\_ip.txt 192.168.0.2 tftpboot/hb\_new/IP 2

load\_hb\_pkt hb\_64\_dos.txt 192.168.0.2 tftpboot 1

load\_hb\_pkt hb\_64\_unix.txt 192.168.0.2 tftpboot 1

load\_hb\_pkt hb\_64.bin 192.168.0.2 tftpboot 1